

EXECUTIVE PROCESS MONITORING OF WORKER PROCESSES

BACKGROUND OF THE INVENTION

Technical Field

This invention relates generally to worker processes that may be improperly or maliciously functioning, and more particularly to monitoring such worker processes by an executive process.

Description of the Prior Art

Modern computer systems typically have firmware, or other non-volatile memory. Firmware is generally a category of memory chips that hold their content without electrical power and include read-only memory (ROM), programmable ROM (PROM), erasable and programmable ROM (EPROM) and electrically erasable and programmable ROM (EEPROM) technologies. Firmware becomes “hard software” when holding program code. For example, in some computer systems, the firmware may include the basic input/output system (BIOS) of a system. The BIOS is a set of routines in a computer, which is stored on a chip and provides an interface between the operating system and the hardware. The BIOS supports all peripheral technologies and internal services, such as the real-time clock.

Within firmware and other computer operating environments, there generally exists a limited-functionality operating mode for performing hardware diagnostic testing, which is also known as performing diagnostics. Hardware diagnostic environments customarily are single-threaded environments without interrupt support, which means that the running process has complete control within the environment. Other processes

8

9

10

11

2

terminating the process in response to determining that the process is improperly functioning, and for returning the processor control back to the process in response to determining that the process is properly functioning.

12 A system of the invention includes an executive process and a worker process. The executive process has a heartbeat interface. The worker process periodically calls the heartbeat interface so that the executive process is able to determine whether the worker process is improperly functioning. Other features and advantages of the invention will become apparent from the following detailed description of the presently preferred embodiment of the invention, taken in conjunction with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

13 FIG. 1 is a flowchart of a method for an executive process to monitor a worker process, according a preferred embodiment of the invention, and is suggested for printing on the first page of the patent.

14 FIG. 2 is a diagram of a system architecture having firmware in conjunction with which embodiments of the invention may be implemented.

15 FIG. 3 is a diagram of another system architecture having firmware, that can be consistent with the system architecture of FIG. 2, in conjunction with which embodiments of the invention may be implemented.

16 FIG. 4 is a diagram of an operating environment showing the functioning of an embodiment of the invention in which a worker process calls a heartbeat application programming interface (API) of an executive process, so the executive process is able to monitor functioning of the worker process. The embodiment of FIG. 4 is consistent with the embodiment of FIG. 1.

FIG. 5 is a flowchart of a method showing in more detail how an embodiment of the invention operates, such that an executive process is able to monitor functioning of a worker process. The embodiment of FIG. 5 is consistent with the embodiments of FIGs. 1 and 4.

DESCRIPTION OF THE PREFERRED EMBODIMENT

Overview

FIG. 1 shows a method 100 according to an embodiment of the invention. Like other methods of the invention, the method 100 can be implemented as means stored on a computer-readable medium of an article of manufacture. The computer-readable medium may be a recordable data storage medium, a modulated carrier signal, or another type of computer-readable medium.

The method 100 is performed by an executive process and a worker process, and can be performed within a firmware operating environment or another type of operating environment. An executive process is a process that has higher priority than a worker process, and thus a worker process is a process that has lower priority than an executive process. The operating environment may be a single-threaded environment that does not support interrupts, such that a process must willingly transfer processor control to another process. The worker process may be a diagnostic test process for the firmware, or another type of process.

Periodically, therefore, the worker process sends a signal to the executive process (102). The sending of this signal may be the calling of a heartbeat application programming interface (API) of the executive process by the worker process. More generally, the heartbeat API is a heartbeat interface of the executive process. By sending

this signal to the executive process, the worker process transfers processor control to the executive process. The executive process thus receives the signal from the worker process (104), and has processor control, such that it can execute within the operating environment.

24

The executive process determines whether the worker process is improperly functioning (106). The worker process may be properly functioning, or it may be improperly functioning in a number of different manners. For example, the worker process may simply be malfunctioning, such that it is not consciously functioning as it should. As another example, the worker process may be maliciously functioning, such that it is consciously not functioning as it should.

25

In response to determining that the worker process is improperly functioning (108), the executive process terminates the worker process (112). Before or after terminating the worker process, the executive process may also perform diagnostic functionality aimed at assisting programmers to understand why the worker process was not properly functioning. For instance, the executive process may record a copy of the worker process in a storage, so that the worker process may be later examined.

26

If the worker process is functioning properly (108), the executive process ultimately returns processor control back to the worker process (110), such that the worker process again periodically sends a signal to the executive process in the future. The executive process may also increment a hardware heartbeat counter for the worker process. The executive process may also perform other functionality.

Technical Background

FIG. 2 shows an example architecture 200 in conjunction with which embodiments of the invention may be implemented. The architecture 200 is specifically that of the Intel Architecture (IA), which includes ITANIUM processors manufactured by Intel Corp., of Santa Clara, Calif. The invention itself may be implemented in conjunction with other architectures, however. The architecture 200 includes platform hardware 202 other than the processor 220, a processor abstraction layer (PAL) 210, a system abstraction layer (SAL) 212, and operating system (OS) software 216. The platform hardware 202 is the hardware on which the operating system 216 is running, except for the processor 220. The PAL 210 provides a consistent firmware interface to abstract processor implementation-specific features to the other components of the architecture 200. The SAL 212 isolates operating system and other higher-level software from implementation differences in the platform hardware 202.

The architecture 200 also includes a legacy OS loader 204, a legacy basic input/output system (BIOS) 206, an Extensible Firmware Interface (EFI) OS loader 208, EFI boot services 214, and EFI runtime services 218. The Extensible Firmware Interface (EFI) is that which has been promulgated by Intel Corp., of Santa Clara, Calif. The legacy OS loader 204 is present for the architecture 200 to be compatible with older computer systems that still include such a loader 204 for loading the OS 216. Similarly, the legacy BIOS 206 is present for the architecture 200 to be compatible with older computer systems that still include such BIOS 216. The EFI boot services 214 are the services that the EFI provides for booting the computer of which the architecture 200 is a

part, whereas the EFI runtime services 218 are the services that the EFI provides after the OS software 216 has been loaded.

FIG. 3 shows another view of an architecture 300 in conjunction with which embodiments of the invention can be implemented. The architecture 300 includes platform hardware 302, on top of which hardware-specific BIOS 304 is present, and that interacts with the hardware 302. The hardware 302 may be the hardware 202 of FIG. 2. The BIOS 304 may be the legacy BIOS 206 of FIG. 2. The EFI 306 thus abstracts the BIOS 304 from the OS, decoupling development. The EFI 306 is a polling-based, single-threaded firmware environment, which in a diagnostic mode does not support interrupts. The EFI 306 may be inclusive of the EFI boot services 214, the EFI runtime services 218, the SAL 212, and the PAL 210 of FIG. 2.

The EFI 306 allows for the creation of programs and driver programs. The former are resident in memory only while they execute, whereas the latter remain resident in memory until removed. More specifically, there are two types of driver programs: boot services drivers and runtime services drivers. The boot services drivers cease to exist when a call is made to an exit boot services API of the EFI, which is typically made by the OS loader, such as the OS loader 208 of FIG. 2. Runtime services drivers remain resident while the OS is running.

Executive Process Monitoring of Worker Process

FIG. 4 shows the execution of an executive process 402 and a worker process 404 according to an embodiment of the invention. The execution is preferably within an operating environment afforded by the EFI 306 in at least a diagnostic mode of the EFI 306, as has been described. The executive process 402 and the worker process 404 may

each be a driver program within the EFI 306, such that they remain resident in memory even when they are not functioning and do not have active processor control.

Periodically, the worker process 404 calls a heartbeat API 406 of the executive process 402, as indicated by the arrow 408. The executive process 402 can then diagnose the worker process 404 to ensure that it is properly functioning. If it is, then the executive process 402 increments the hardware counter 412 for the worker process 404, as indicated by the arrow 410. That is, the API 406 of the executive processor 402 is inserted between the worker process 404 and the hardware counter 412, such that the worker process 404 periodically calls the API 406, in lieu of periodically incrementing the hardware counter 412.

Because the operating environment may be a single-threaded, interrupt-free environment, as has been described, the executive process 402 in such instances receives processor control from the worker process 404 only when the latter calls the API 406. That is, the executive process 402 may not be able to interrupt running of the worker process 404. However, at least while the worker process 404 is sufficiently functioning properly to call the API 406 of the executive process 402, the executive process 402 is able to monitor the functioning of the worker process 404. The executive process 402 may determine, for instance, that the worker process 404 is about to malfunction, such that it can be said that the worker processor 404 is not properly functioning. The executive process 402 may then terminate the worker process 404, and perform diagnostic measures relative to the worker process 404 as appropriate. If the worker process 404 is functioning, however, the executive process 402 returns process control back to the worker process 404.

35

FIG. 5 shows a method 500 that illustrates in detail the operation of an embodiment of the invention. The method 500 is consistent with the embodiments of FIG. 1 and FIG. 4 that have been described. A firmware diagnostic-mode operating environment is first initialized (502), and previously set commands indicating the type of diagnostic tests that should be run are processed by an executive process (504). The commands may indicate, for instance, which worker processes should be loaded and run to perform diagnostic tests. Thus, a worker process is loaded (506), and runs (508). Periodically, the worker process calls a heartbeat API of the executive process (510), which transfers process control to the executive process. The executive process increments the hardware heartbeat counter for the worker process (512), and then determines whether the worker process should be removed, or terminated (514).

36

If the worker process is properly functioning (514), then the worker process is not removed, and processor control reverts back to the worker process from the executive process. If the worker process is not finished with the diagnostic tests or other functionality (516), then it continues running (508). If the worker process is finished (516), however, then processor control again reverts back to the executive process. As indicated as the junction 517, the executive process either processes more commands in 504, or is finished, and exits the method 500 (524). If, however, the worker process is not properly functioning (514), then the worker process is removed or terminated by the executive process (518), and preferably the executive process diagnoses the reason why the worker process improperly functioned (520). As indicated as the junction 522, the executive process then either processes more commands in 504, or is finished, and exits the method 500 (524).

Advantages over the Prior Art

Embodiments of the invention allow for advantages over the prior art. Unlike the hardware counter-incrementation approach of the prior art, which does not provide for any diagnosis as to why a process is malfunctioning, the heartbeat API approach of the invention provides for such diagnosis. An executive process, when periodically called by a worker process, is able to assess whether the worker process is functioning properly. If the worker process is not functioning properly, the executive process can undertake a diagnosis as to why the worker process is not properly functioning, instead of just resetting the system, as may be the only option available within the prior art.

Alternative Embodiments

It will be appreciated that, although specific embodiments of the invention have been described herein for purposes of illustration, various modifications may be made without departing from the spirit and scope of the invention. For example, embodiments of the invention have been substantially described in relation to a firmware diagnostic-mode operating environment that is single-threaded and does not support interrupts. However, the invention may also be implemented in conjunction with other types of operating environments, including other types of firmware operating environments. Accordingly, the scope of protection of this invention is limited only by the following claims and their equivalents.